CS595 Interative Machine Learning – Project Report Rohan Digambar Gawade
A20379951

BestBuy Tweet Sentiment Analysis – Learning with Rationale Framework

Abstract: - Sentiment is a feeling or emotion. With the advent of world wide web and social media like Facebook, twitter, etc. people have started expressing their emotions through this new medium of expressing their emotions. The large amount of data is being generated by the social media, in the form of tweets, blogs, status updates etc. Knowing the customer opinions by analyzing the customer sentiments is very useful. This report is the study of how Learning with Rationale model can be implemented to classify the tweets into Positive and Negative class and see the effect of this framework on the accuracy of the classifier.

Introduction

Learning with rationale is a framework where the rationales are provided to the documents and label by the labeler and a labeled dataset is created. This dataset can be used by different classifiers for building a model. I.e. there is no need to build the model specific dataset. I am using this framework on BestBuy tweets collected using Twitter Search API.

Sentiment Analysis is the domain I am using for implementing learning with rationale framework. The sentiment analysis helps in knowing the attitude of the people towards the product, topic or service. I am doing sentiment analysis for the tweets.

We can see the example of the movie review for sentiment analysis.

The first review is positive which says the movie is great, the second review talks about the actor which stars the movie and not about the movie hence it is neutral, and the last review says the movie is horrible which is negative review.

Like the movie reviews we are going to predict the sentiments for tweets using the normal model and using learning with rational framework.

CS595 Interative Machine Learning – Project Report Rohan Digambar Gawade
A20379951

Dataset

I collected around 10000 tweets for @BestBuy, which as Text as a column. I used the retrieval code from the GitHub which is available using the following link.

https://github.com/Jefferson-Henrique/GetOldTweets-python

This code fetches the tweets along with date, user handle, who all are addressed and mentioned in the tweets. I modified it to get just the tweet and stored the data in the csv file. This is all raw data with no labels. I manually provided the label for positive and negative tweets. I labelled 100 positive tweets and 100 negative tweets and gave rationales for these tweets.

The other tweets are provided with the rationales and labels using these tweets.

Providing Labels and Rationales

The labels and rationale to the rest of the tweets are provided using either of the two approaches
1. Using simulator – Chi Square to provide the rationales.
2. To ask the human labeler to provide the rationale.

Chi- Square Statistics

The Chi-Square statistics is used to find the rationale. The chi-square value of the rationale determines whether the tweet is positive or negative and also uses the word as rationale.
If there are more than one words which are positive and negative in the tweet, then I am summing the positive and negative values of all the words and taking the greater value to indicate the positive (negative) tweet. Now this tweet with highest positive (negative) rationale list is used as rationale to the tweet. I am using this in my experiment.

Human Labeler

Asking the human labeler to provide the rationale along with label is time consuming so I used chi-square statistics for getting the rationales. I have also provided a provision in case if user wants to label the tweet and provide the rationale by itself. The UI for that is as shown in the below image.

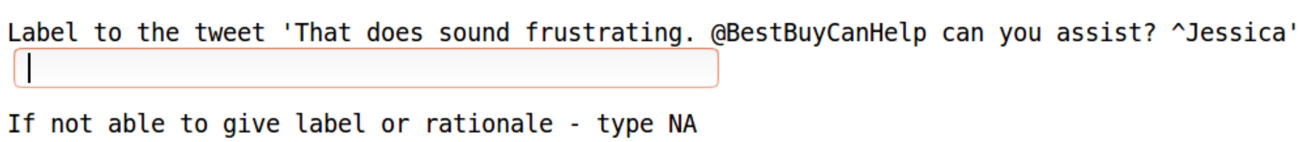


Fig 1

```
If not able to give label or rationale - type NA
Label to the tweet 'That does sound frustrating. @BestBuyCanHelp can you assist? ^Jessica'Negative

Rationale to the tweet 'That does sound frustrating. @BestBuyCanHelp can you assist? ^Jessica'
 frustrating
```

Fig 2

```
If not able to give label or rationale - type NA
Label to the tweet 'That does sound frustrating. @BestBuyCanHelp can you assist? ^Jessica'Negative
Rationale to the tweet 'That does sound frustrating. @BestBuyCanHelp can you assist? ^Jessica'frustrating
Already exist
Negative
frustrating
```

Fig 3

The human labeler is asked to label the tweet and provide the rationale. Now if the rationale provided by the labeler is already present in the list of the rationale it returns as 'the rationale already exists' in the list.

Feature Engineering

Feature is the information that is used for predicting the target. Feature Engineering is the process of creating features that will make machine learning algorithm work. For this experiment, I used Bag of Words model (Count Vectorizer) and Tfidf-matrix (TfidfVectorizer).

Count Vectorizer

It counts the number of times the word occurs in the sentence/tweet/document. The features(words) and their counts are stored in the form of csr matrix

Tfidf Vectorizer

The tfidf representation indicates how important the word is in the document corpus, in our case the tweet corpus. The weight of the word increases proportionally according to the number of times the word appears in the corpus.
Both the representations are as follows:

```
In [13]: textline = ["And worst customer service award goes to @BestBuy WILL NEVER SHOP HERE EVER AGAIN",
                    "Even better when I applied my $10 off reward."]

In [11]: cv = CountVectorizer()
         X = cv.fit_transform(textline)
         print(cv.get_feature_names())
         X.toarray()

         ['10', 'again', 'and', 'applied', 'award', 'bestbuy', 'better', 'customer', 'even', 'ever', 'goes', 'here', 'my', 'ne
         ver', 'off', 'reward', 'service', 'shop', 'to', 'when', 'will', 'worst']

Out[11]: array([[0, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1],
                [1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0]], dtype=int64)

In [12]: tv = TfidfVectorizer()
         X = tv.fit_transform(textline)
         print(tv.get_feature_names())
         X.toarray()

         ['10', 'again', 'and', 'applied', 'award', 'bestbuy', 'better', 'customer', 'even', 'ever', 'goes', 'here', 'my', 'ne
         ver', 'off', 'reward', 'service', 'shop', 'to', 'when', 'will', 'worst']

Out[12]: array([[ 0.        ,  0.26726124,  0.26726124,  0.        ,  0.26726124,
                  0.26726124,  0.        ,  0.26726124,  0.        ,  0.26726124,
                  0.26726124,  0.26726124,  0.        ,  0.26726124,  0.        ,
                  0.        ,  0.26726124,  0.26726124,  0.26726124,  0.        ,
                  0.26726124,  0.26726124],
                [ 0.35355339,  0.        ,  0.        ,  0.35355339,  0.        ,
                  0.        ,  0.35355339,  0.        ,  0.35355339,  0.        ,
                  0.        ,  0.        ,  0.35355339,  0.        ,  0.35355339,
                  0.35355339,  0.        ,  0.        ,  0.        ,  0.35355339,
                  0.        ,  0.        ]])
```

The first one which contains 1 and 0 value is the Count Vectorizer representation. The one below it is a Tfidf Representation.

Learning with Rationale

I am using the learning with rationale framework wherein I multiply the rationale in the tweet with 1 and words other than rationale with 0.1 or 0.01. Thus, the words that are rationale are multiplied with higher weight indicating the importance of the words.
Example as per the research paper. “Active Learning with Rationales for Text Classification “

	great	movie	plot	performance	actor	terrible	avoid	outdoor	cinema	atmosphere	terrific
Lw/oR Representation (binary)											
D1	1	1									
D2	1		1	1	1	1	1				
D3	1	1						1	1	1	1
LwR Transformation of the binary Lw/oR repr.											
D1	*r*	*o*									
D2	*o*		*o*	*o*	*o*	*r*	*r*				
D3	*o*	*o*						*o*	*o*	*o*	*r*

I edited the CountVectorizer() and TfidfVectorizer() returned csr matrix and multiplied the rationale in the tweets with r and o values.
The representation is as below:

CountVectorizer() with Rationale

```
['10', 'again', 'and', 'applied', 'award', 'bestbuy', 'better', 'customer', 'even', 'ever', 'goes', 'here', 'my', 'ne
ver', 'off', 'reward', 'service', 'shop', 'to', 'when', 'will', 'worst']
[[ 0.    0.01  0.01  0.    0.01  0.01  0.    0.01  0.    0.01  0.01  0.01
   0.    0.01  0.    0.    0.01  0.01  0.01  0.    0.01  1.  ]
 [ 0.01  0.    0.    0.01  0.    0.    1.    0.    0.01  0.    0.    0.
   0.01  0.    0.01  0.01  0.    0.    0.    0.01  0.    0.  ]]
<class 'scipy.sparse.csr.csr_matrix'>
```

TfidfVectorizer() with Rationale

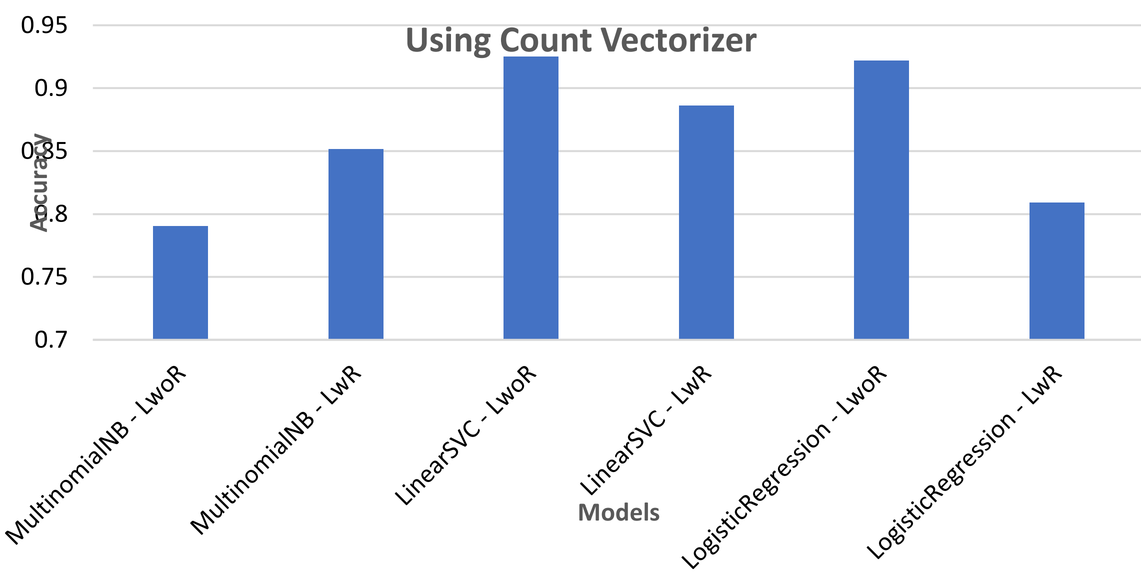
```
['10', 'again', 'and', 'applied', 'award', 'bestbuy', 'better', 'customer', 'even', 'ever', 'goes', 'here', 'my', 'ne
ver', 'off', 'reward', 'service', 'shop', 'to', 'when', 'will', 'worst']
[[ 0.          0.00267261  0.00267261  0.          0.00267261  0.00267261
   0.          0.00267261  0.          0.00267261  0.00267261  0.00267261
   0.          0.00267261  0.          0.          0.00267261  0.00267261
   0.00267261  0.          0.00267261  0.26726124]
 [ 0.00353553  0.          0.          0.00353553  0.          0.
   0.35355339  0.          0.00353553  0.          0.          0.
   0.00353553  0.          0.00353553  0.00353553  0.          0.          0.
   0.00353553  0.          0.        ]]
<class 'scipy.sparse.csr.csr_matrix'>
```

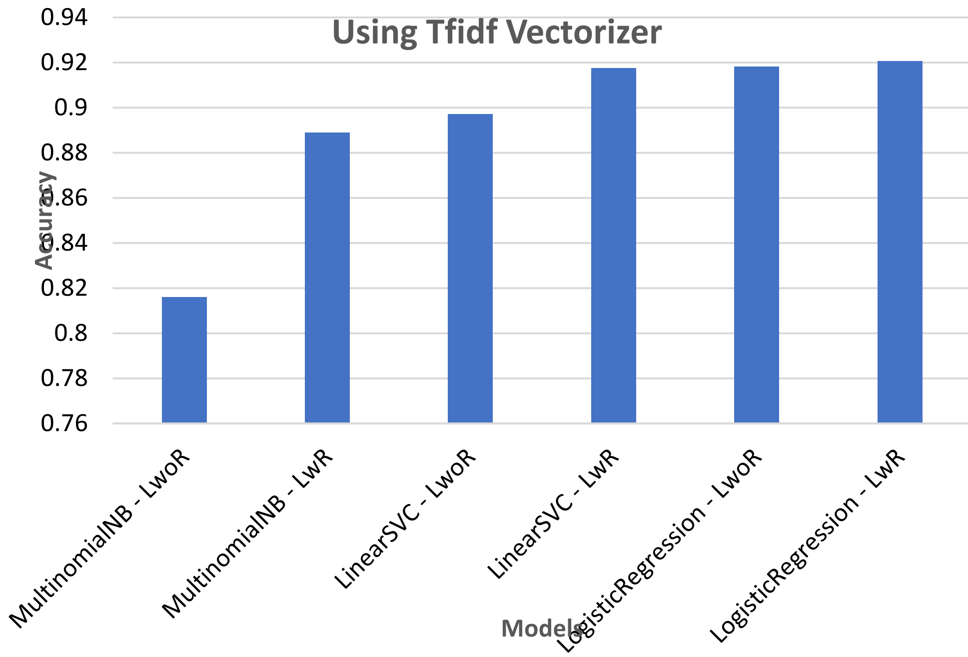
As we can see the in CountVectorizer() and TfidfVectorizer() with rationale the word 'worst' and 'better' are multiplied with r = 1 while other words are multiplied with o = 0.01.

Experiment

The experiment was performed to compare the Learning without Rationale and Learning with Rationale. I used Multinomial Naive Bayes, Logistic Regression and Support Vector Machines models to carry out the experiment.
I took the value r = 1 and o = 0.1 and calculated the accuracy for all the model with LwoR and LwR configuration.

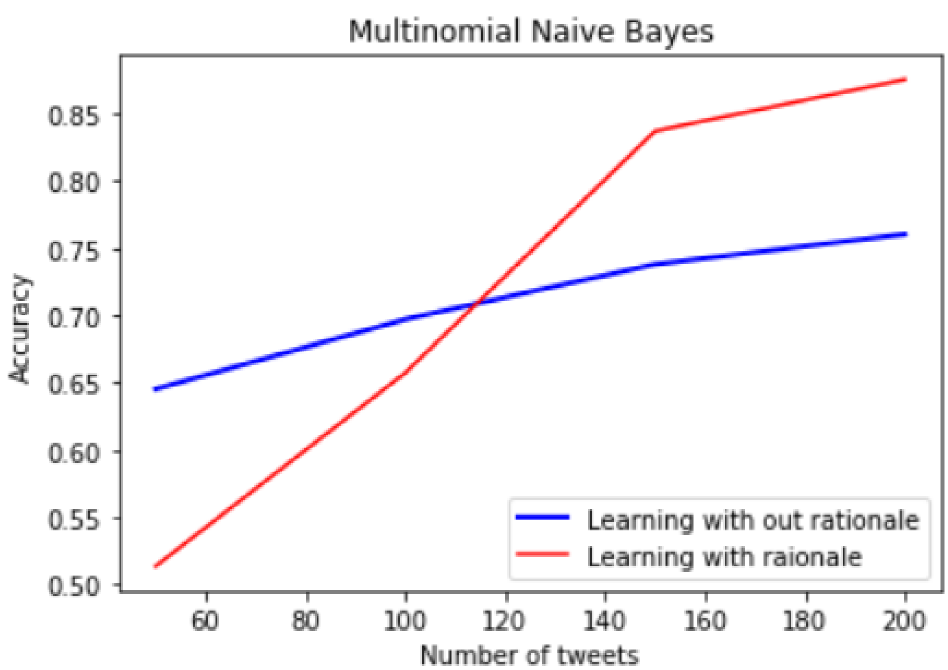
The results are as follows.

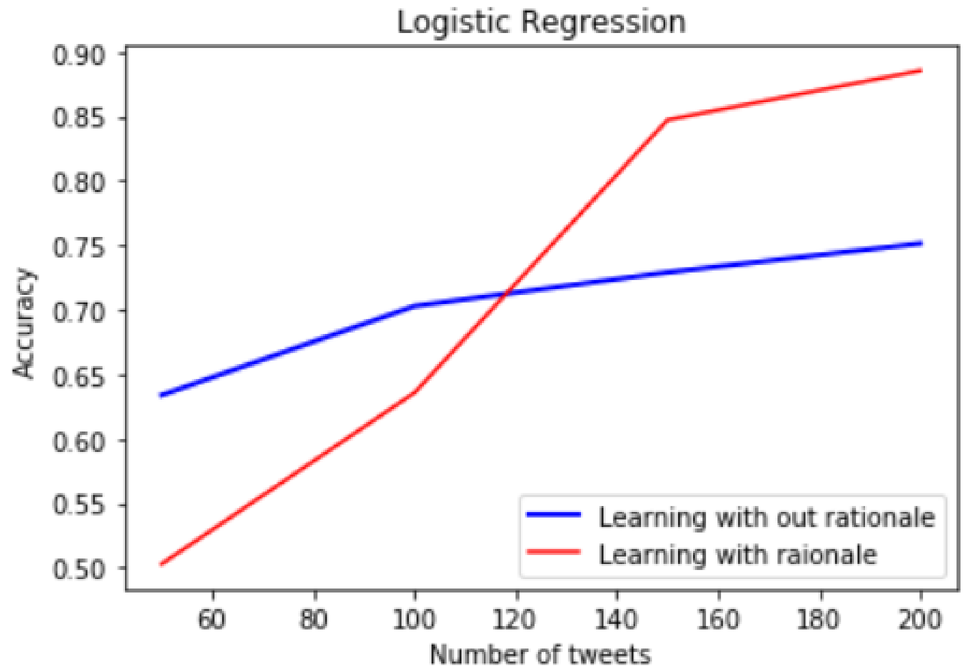


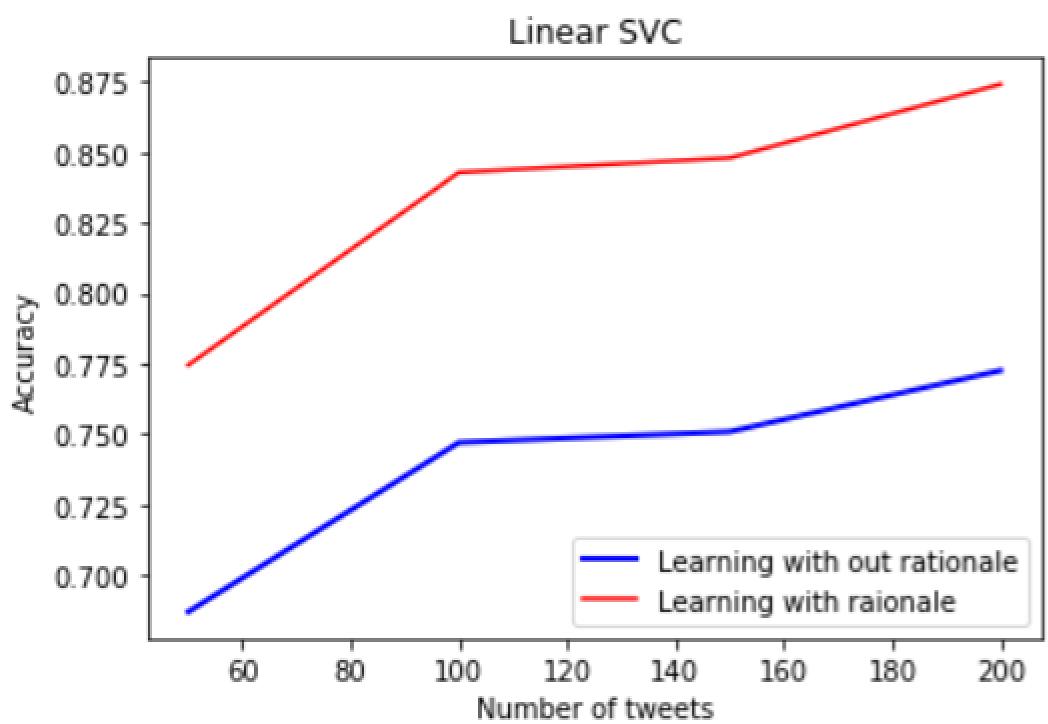


The MultinomialNB worked well for both the text representation but LinearSVC and Logistic Regression accuracy came down when used with LwR framework for count vectorizer representation. There was significant improvement for Tfidf representation for all the three models over LwoR.

Then I changed the value of o = 0.01.
And plotted the graph for LwoR vs LwR for all three models taking 50,100,150, and 200 tweets for training the models and testing on the data.
The graphs are as follows







The graph for multinomial naïve bayes and logistic regression doesnt follow as per mentioned in the research paper. The learning with rationale initially gives better result compared to LwoR as per the paper. But here we get reverse output. The two model learns better as per the training set increases.
The Linear SVC follows the research paper output. The LwR accuracy starts better than LwOR and increases as the number of training set increases.

Conclusion

The accuracy provided by LwR was better as compared to LwoR for all the three models when considered with whole dataset. The LwR approached worked well with IMDB and NewsGroup dataset. May with the twitter data we need to have more tuned tweets as the data from the twitter is not proper and also contains lot of unwanted words, hyperlink etc. I performed this experiment without doing text preprocessing like not removing stop words, stemming, removing links which could have provided a different result. But eventually all the three models provided better results as they learned more once the training set grew. In this approach I have just used 1 gram, in future I proposed to use 2-grams or 3-grams. Giving more weights to the rationale actually makes sense as we are providing higher weight to the important feature. The models should actually perform well when we have less training examples.

References

1. Active Learning with Rationales for Text Classification
http://www.cs.iit.edu/~ml/pdfs/sharma-naaclhlt15.pdf

2. http://scikit-learn.org/stable/